

Perancangan Dasbor Inspiscity – Menelusuri Model Ketimpangan Sosial Ekonomi Indonesia

Ferdian Saputra (2KS3)

2025-07-23

Pendahuluan

Ketimpangan sosial ekonomi merupakan isu sentral dalam pembangunan di Indonesia. Meskipun berbagai upaya telah dilakukan untuk menurunkan angka kemiskinan dan meningkatkan kesejahteraan, disparitas antar wilayah — baik antar provinsi maupun kabupaten/kota — masih sangat terasa. Ketimpangan ini tidak hanya tercermin dari tingkat kemiskinan, tetapi juga dari akses pendidikan, infrastruktur dasar, kepemilikan rumah, hingga kerentanan terhadap bencana. Oleh karena itu, pemetaan dan analisis ketimpangan sosial ekonomi menjadi sangat penting untuk mendukung perumusan kebijakan yang lebih adil dan tepat sasaran.

Dasbor “Inspiscity” hadir sebagai alat bantu visualisasi dan analisis interaktif untuk memahami pola ketimpangan sosial ekonomi di Indonesia. Dengan mengintegrasikan data numerik dan spasial, dasbor ini memungkinkan pengguna untuk mengeksplorasi berbagai indikator sosial-ekonomi secara komprehensif dan mudah dipahami.

Data & Variabel yang Digunakan

Data utama yang digunakan dalam dasbor ini berasal dari file “UAS-Data+GIS.xlsx” (data sosial ekonomi kabupaten/kota seluruh Indonesia) dan “indonesia511.geojson” (batas wilayah spasial). Data ini merupakan hasil ekspor dan pengolahan dari QGIS, sehingga sudah terintegrasi antara data numerik dan spasial. Variabel-variabel yang dianalisis antara lain:

- **POVERTY** (Kemiskinan): Persentase penduduk miskin, sebagai indikator utama ketimpangan.
- **LOWEDU** (Pendidikan Rendah): Persentase penduduk usia 15 tahun ke atas berpendidikan rendah, menggambarkan akses pendidikan.
- **ILLITERATE** (Buta Huruf): Persentase penduduk yang tidak bisa membaca dan menulis, sebagai indikator kualitas SDM.
- **CHILDREN** (Anak Balita): Persentase penduduk usia di bawah lima tahun, terkait beban tanggungan keluarga.
- **ELDERLY** (Lansia): Persentase penduduk usia 65 tahun ke atas, terkait beban sosial dan kesehatan.
- **NOELECTRIC** (Rumah Tangga Non-Listrik): Persentase rumah tangga tanpa listrik, indikator infrastruktur dasar.
- **NOTRAINING** (Tidak Dapat Pelatihan): Persentase rumah tangga yang tidak mendapat pelatihan bencana, terkait resiliensi sosial.
- **DPRONE** (Rawan Bencana): Persentase rumah tangga di daerah rawan bencana.
- **RENTED** (Sewa Rumah): Persentase rumah tangga yang menyewa rumah, indikator kepemilikan aset.
- **NOSEWER** (Tidak Ada Drainase): Persentase rumah tangga tanpa sistem drainase.
- **TAPWATER** (Air Ledeng): Persentase rumah tangga yang menggunakan air ledeng.
- **POPULATION** (Jumlah Penduduk): Jumlah total penduduk.

Hubungan antar variabel ini sangat erat. Misalnya, kemiskinan seringkali berkorelasi dengan rendahnya pendidikan, akses infrastruktur, dan kerentanan terhadap bencana. Dengan menganalisis variabel-variabel ini secara spasial dan statistik, kita dapat mengidentifikasi wilayah yang paling tertinggal dan faktor-faktor utama penyebab ketimpangan.

Fitur & Alur Dasbor Inspiscity

Dasbor ini dirancang agar mudah digunakan dan informatif. Pengguna dapat:

- **Manajemen Data:** Melakukan kategorisasi variabel numerik menjadi kategori (rendah, sedang, tinggi), melihat distribusi kategori, dan mengunduh data hasil kategorisasi.

- Eksplorasi Data: Melihat grafik batang rata-rata per provinsi/kabupaten, serta peta interaktif yang mewarnai wilayah sesuai nilai variabel yang dipilih. Interpretasi otomatis membantu memahami pola mayoritas, nilai minimum/maksimum, dan outlier.
- Uji Asumsi: Melakukan uji normalitas (Shapiro-Wilk) dan homogenitas (Levene) untuk memastikan data memenuhi syarat analisis statistik lanjut. Hasil uji menampilkan hipotesis awal (H_0), nilai statistik uji, p-value, dan keputusan (Tolak H_0 /Gagal Tolak H_0).
- Statistik Inferensia: Melakukan uji beda rata-rata, proporsi, varians, dan ANOVA, lengkap dengan interpretasi otomatis, hipotesis, nilai statistik uji, p-value, dan keputusan.
- Regresi Linier Berganda: Membangun model regresi dari variabel kontrol yang dipilih user, menampilkan interpretasi koefisien, signifikansi, serta hasil uji asumsi klasik (normalitas residual, multikolinearitas, heteroskedastisitas) beserta hipotesis dan keputusan. Model terbaik otomatis (stepwise regression) juga ditampilkan dari variabel yang dipilih.

Pembuatan Kode Program

Untuk merancang dasbor ini, perlu kode program yang disusun menggunakan bahasa R dengan beberapa library, sehingga memungkinkan dasbor terwujud untuk menyediakan hasil visualisasi data dengan berbagai macam bentuk. Hasil output dari dasbor yang digunakan oleh pengguna nantinya dapat dilihat sebagai contoh pada folder yang terlampir yang masih menjadi satu folder juga dengan laporan maupun program ini. Berikut kode program yang dijalankan dalam penyusunan dasbor Inspiciency.

1. Menentukan Daftar Library

```
# app.R
# Dasbor RShiny: "Inspiciency"

# Library
library(shiny)
library(shinydashboard)
##
## Attaching package: 'shinydashboard'
```

```
## The following object is masked from 'package:graphics':
##
##   box
library(bslib)
##
## Attaching package: 'bslib'
## The following object is masked from 'package:utils':
##
##   page
library(DT)
##
## Attaching package: 'DT'
## The following objects are masked from 'package:shiny':
##
##   dataTableOutput, renderDataTable
library(ggplot2)
library(readxl)
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
library(sf)
## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf_use_s2() is TRUE
library(leaflet)
library(shinyjs)
##
## Attaching package: 'shinyjs'
```

```
## The following object is masked from 'package:shiny':
##
##   runExample
## The following objects are masked from 'package:methods':
##
##   removeClass, show
library(shinyWidgets)
##
## Attaching package: 'shinyWidgets'
## The following object is masked from 'package:shinyjs':
##
##   alert
library(car)
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##   recode
library(plotly)
##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##   last_plot
## The following object is masked from 'package:stats':
##
##   filter
## The following object is masked from 'package:graphics':
##
##   layout
library(shinycssloaders)
library(lmtest)
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
##
## Attaching package: 'lmtest'
## The following object is masked from 'package:shinyjs':
##
##   reset
```

2. Mengimpor Sumber Data & Konversi Tipe Data

```
# Data Utama
path_data <- "UAS-Data+GIS.xlsx"
path_geojson <- "indonesia511.geojson"
data_raw <- readxl::read_excel(path_data, sheet = 1)
geojson <- sf::st_read(path_geojson, quiet = TRUE)

# Konversi kolom join dan label wilayah ke character dan trim spasi
for (col in c("kodeprkab", "kdprov", "kdkab", "nmprov", "nmkab")) {
  if (col %in% colnames(data_raw)) data_raw[[col]] <- trimws(as.character(data_raw[[col]]))
  if (col %in% colnames(geojson)) geojson[[col]] <- trimws(as.character(geojson[[col]]))
}

# Samakan format digit kode wilayah (misal kodeprkab 4 digit, kdprov 2 digit, kdkab 2 digit).
if ("kodeprkab" %in% colnames(data_raw)) data_raw[["kodeprkab"]] <- sprintf("%04s",
data_raw[["kodeprkab"]])
if ("kodeprkab" %in% colnames(geojson)) geojson[["kodeprkab"]] <- sprintf("%04s",
geojson[["kodeprkab"]])
if ("kdprov" %in% colnames(data_raw)) data_raw[["kdprov"]] <- sprintf("%02s",
data_raw[["kdprov"]])
if ("kdprov" %in% colnames(geojson)) geojson[["kdprov"]] <- sprintf("%02s", geojson[["kdprov"]])
if ("kdkab" %in% colnames(data_raw)) data_raw[["kdkab"]] <- sprintf("%02s", data_raw[["kdkab"]])
if ("kdkab" %in% colnames(geojson)) geojson[["kdkab"]] <- sprintf("%02s", geojson[["kdkab"]])
```

3. Mendefinisikan Variabel dalam Data Impor

```
prov_col <- "nmprov"
kab_col <- "nmkab"
kdprov_col <- "kdprov"
kodeprkab_col <- "kodeprkab"
var_respon <- "POVERTY"
var_bebas <- c(
  "CHILDREN", "FEMALE", "ELDERLY", "FHEAD", "FAMILYSIZE",
  "NOELECTRIC", "LOWEDU", "GROWTH", "POVERTY", "ILLITERATE", "NOTRAINING",
  "DPRONE", "RENTED", "NOSEWER", "TAPWATER", "POPULATION"
)
var_numerik <- c(var_respon, var_bebas)
```

4. Mengkategorisir Nilai Variabel

```
make_kategori_label <- function(x, n = 3) {
  if (n == 3) return(c("Rendah", "Sedang", "Tinggi"))
  if (n == 4) return(c("Sangat Rendah", "Rendah", "Sedang", "Tinggi"))
  return(paste("Kategori", 1:n))
}
make_kategori <- function(x, n = 3, label = NULL) {
  if (is.numeric(x)) {
    if (is.null(label)) label <- make_kategori_label(x, n)
    cut(x, breaks = n, labels = label, include.lowest = TRUE)
  } else {
    as.factor(x)
  }
}
```

5. Membuat Desain Antarmuka Pengguna

```
# UI
ui <- dashboardPage(
  skin = "blue",
  dashboardHeader(title = "Insciety Dashboard"),
```

```

dashboardSidebar(
  sidebarMenu(
    menuItem("Beranda", tabName = "beranda", icon = icon("home")),
    menuItem("Manajemen Data", tabName = "manajemen", icon = icon("database")),
    menuItem("Eksplorasi Data", tabName = "eksplorasi", icon = icon("chart-bar")),
    menuItem("Uji Asumsi", tabName = "asumsi", icon = icon("check-circle")),
    menuItem("Statistik Inferensia", icon = icon("flask"),
      menuSubItem("Uji Beda Rata-rata", tabName = "beda_rata"),
      menuSubItem("Uji Proporsi", tabName = "uji_proporsi"),
      menuSubItem("Uji Varians", tabName = "uji_varians"),
      menuSubItem("ANOVA", tabName = "anova")
    ),
    menuItem("Regresi Linier Berganda", tabName = "regresi", icon = icon("project-diagram"))
  )
),
dashboardBody(
  useShinyjs(),
  tags$head(tags$style(HTML(".interpretasi-box {background: #f8f8f8; border: 1px solid #ccc; padding: 8px; margin-top: 8px; min-height: 40px; font-family: monospace; font-size: 15px;} .shiny-output-error-validation {color: red;} .shiny-text-output {overflow-x: hidden !important;}"))),
  tabItems(
    tabItem(tabName = "beranda",
      h2("Inspiciety: Inspect Indonesian Socio-Economic Inequality"),
      tags$hr(),
      h3("Topik Dasbor"),
      p("Dasbor ini membahas Ketimpangan Ekonomi Wilayah di Indonesia. Variabel respon utama: POVERTY (kemiskinan). Variabel kontrol: indikator sosial-ekonomi lain (lihat Metadata di bawah)."),
      h3("Sumber Data"),
      tags$ul(
        tags$li("UAS-Data+GIS.xlsx (data utama kabupaten/kota)"),
        tags$li("UAS-Matrik_Penimbang_Jarak.xlsx (matriks jarak/spasial)"),
        tags$li("indonesia511.geojson (peta wilayah)"),
        tags$li("Metadata.pdf")
      ),
    ),
  )
)

```



```

h3("Fitur Dasbor"),
tags$Sul(
  tags$Li("Manajemen Data: kategorisasi variabel kontinu menjadi kategorik, interpretasi
otomatis, unduh hasil."),
  tags$Li("Eksplorasi Data: statistik deskriptif, histogram, peta visualisasi, tabel, interpretasi
otomatis."),
  tags$Li("Uji Asumsi: uji normalitas & homogenitas, interpretasi otomatis."),
  tags$Li("Statistik Inferensia: uji beda rata-rata, proporsi, varians, ANOVA, interpretasi
otomatis."),
  tags$Li("Regresi Linier Berganda: model, uji asumsi, interpretasi otomatis.")
),
h3("Informasi Variabel"),
tags$Table(class = "table table-bordered",
  tags$thead(
    tags$tr(
      tags$th("Label"),
      tags$th("Variabel"),
      tags$th("Deskripsi (Bahasa Indonesia)")
    )
  ),
  tags$tbody(
    tags$tr(tags$td("DISTRICTCODE"), tags$td("Kode Wilayah"), tags$td("Kode unik
untuk setiap kabupaten/kota")),
    tags$tr(tags$td("CHILDREN"), tags$td("Anak Balita"), tags$td("Persentase penduduk
usia di bawah lima tahun")),
    tags$tr(tags$td("FEMALE"), tags$td("Perempuan"), tags$td("Persentase penduduk
perempuan")),
    tags$tr(tags$td("ELDERLY"), tags$td("Lansia"), tags$td("Persentase penduduk usia 65
tahun ke atas")),
    tags$tr(tags$td("FHEAD"), tags$td("Kepala Rumah Tangga Perempuan"),
tags$td("Persentase rumah tangga dengan kepala rumah tangga perempuan")),
    tags$tr(tags$td("FAMILYSIZE"), tags$td("Jumlah Anggota Rumah Tangga"),
tags$td("Rata-rata jumlah anggota rumah tangga per kabupaten/kota")),
    tags$tr(tags$td("NOELECTRIC"), tags$td("Rumah Tangga Non-Listrik")),

```

```

tags$td("Persentase rumah tangga yang tidak menggunakan listrik sebagai sumber penerangan")),
      tags$str(tags$td("LOWEDU"), tags$td("Pendidikan Rendah"), tags$td("Persentase
penduduk usia 15 tahun ke atas berpendidikan rendah")),
      tags$str(tags$td("GROWTH"), tags$td("Pertumbuhan Penduduk"), tags$td("Persentase
perubahan jumlah penduduk")),
      tags$str(tags$td("POVERTY"), tags$td("Kemiskinan"), tags$td("Persentase penduduk
miskin")),
      tags$str(tags$td("ILLITERATE"), tags$td("Buta Huruf"), tags$td("Persentase penduduk
yang tidak bisa membaca dan menulis")),
      tags$str(tags$td("NOTRAINING"), tags$td("Tidak Dapat Pelatihan"),
tags$td("Persentase rumah tangga yang tidak mendapat pelatihan bencana")),
      tags$str(tags$td("DPRONE"), tags$td("Rawan Bencana"), tags$td("Persentase rumah
tangga yang tinggal di daerah rawan bencana")),
      tags$str(tags$td("RENTED"), tags$td("Sewa Rumah"), tags$td("Persentase rumah
tangga yang menyewa rumah")),
      tags$str(tags$td("NOSEWER"), tags$td("Tidak Ada Drainase"), tags$td("Persentase
rumah tangga yang tidak memiliki sistem drainase")),
      tags$str(tags$td("TAPWATER"), tags$td("Air Ledeng"), tags$td("Persentase rumah
tangga yang menggunakan air ledeng")),
      tags$str(tags$td("POPULATION"), tags$td("Jumlah Penduduk"), tags$td("Jumlah total
penduduk"))
    )
  )
),
tabItem(tabName = "manajemen",
  h2("Manajemen Data"),
  fluidRow(
    box(width = 4, title = "Filter Wilayah", status = "primary",
      selectInput("prov", "Pilih Provinsi:", choices = c("Semua", unique(data_raw[[prov_col]]))),
      tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu tabel akan muncul."),
    ),
    box(width = 4, title = "Variabel & Kategori", status = "primary",
      selectInput("var_ekonomi", "Pilih Variabel:", choices = var_numerik),

```

```

        selectInput("n_kat", "Jumlah Kategori:", choices = c(3, 4), selected = 3),
        actionButton("buat_kat", "Buat Kategori")
    ),
    box(width = 4, title = "Unduh Data & Interpretasi", status = "primary",
        downloadButton("download_data", "Unduh Data (xlsx)",
            downloadButton("download_interpretasi_kat", "Unduh Interpretasi (Word)",
                downloadButton("download_gabungan_kat", "Unduh Gabungan (Word)")
            )
    ),
    DTOutput("tabel_kat"),
    uiOutput("interpretasi_kat_area")
),
tabItem(tabName = "eksplorasi",
    h2("Eksplorasi Data"),
    fluidRow(
        box(width = 4, title = "Filter Wilayah & Variabel", status = "primary",
            selectInput("prov2", "Pilih Provinsi:", choices = c("Semua",
unique(data_raw[[prov_col]]))),
            selectInput("var_eksplorasi", "Pilih Variabel:", choices = var_numerik),
            tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
        ),
        box(width = 8, title = "Visualisasi", status = "primary",
            plotlyOutput("plot_eksplorasi") %>% withSpinner(),
            DTOutput("tabel_eksplorasi"),
            downloadButton("download_plot_eksplorasi", "Unduh Gambar (PNG)")
        )
    ),
    fluidRow(
        box(width = 12, title = "Peta", status = "primary",
            selectInput("map_var", "Pilih Variabel Peta:", choices = var_numerik, selected =
var_numerik[1]),
            leafletOutput("map_eksplorasi") %>% withSpinner(),
            downloadButton("download_map_eksplorasi", "Unduh Peta (JPG)")
        )
    )
)

```

```

    )
  ),
  uiOutput("interpretasi_eksplorasi_area"),
),
tabItem(tabName = "asumsi",
  h2("Uji Asumsi Data"),
  fluidRow(
    box(width = 4, title = "Filter Wilayah & Variabel", status = "primary",
      selectInput("prov3", "Pilih Provinsi:", choices = c("Semua",
unique(data_raw[[prov_col]]))),
      uiOutput("prov3_2_ui"),
      selectInput("var_asumsi", "Pilih Variabel:", choices = var_numerik),
      tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
    ),
    box(width = 8, title = "Uji Normalitas & Homogenitas", status = "primary",
      plotOutput("plot_norm", height = 250) %>% withSpinner(),
      plotOutput("plot_homog", height = 250) %>% withSpinner(),
      verbatimTextOutput("uji_norm"),
      verbatimTextOutput("uji_homog"),
      downloadButton("download_norm", "Unduh Output Normalitas"),
      downloadButton("download_homog", "Unduh Output Homogenitas")
    )
  ),
  uiOutput("interpretasi_asumsi_area")
),
tabItem(tabName = "beda_rata",
  h2("Uji Beda Rata-rata (1 & 2 Provinsi)"),
  fluidRow(
    box(width = 4, title = "Pengaturan Uji", status = "primary",
      selectInput("var_beda", "Pilih Variabel Respon:", choices = var_numerik),
      radioButtons("kelompok_beda", "Jumlah Kelompok:", choices = c("1 Provinsi", "2
Provinsi")),
      conditionalPanel(

```

```

        condition = "input.kelompok_beda == '1 Provinsi'",
        selectInput("prov_beda1", "Pilih Provinsi:", choices = unique(data_raw[[prov_col]])),
        numericInput("mu0_beda", "Nilai Hipotesis Awal (mu0):", value = 0)
    ),
    conditionalPanel(
        condition = "input.kelompok_beda == '2 Provinsi'",
        selectInput("prov_beda1_2", "Provinsi 1:", choices = unique(data_raw[[prov_col]])),
        selectInput("prov_beda2_2", "Provinsi 2:", choices = unique(data_raw[[prov_col]]))
    ),
    actionButton("uji_beda", "Lakukan Uji"),
    tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
    ),
    box(width = 8, title = "Hasil Uji", status = "primary",
        verbatimTextOutput("output_beda"),
        downloadButton("download_beda", "Unduh Output")
    )
    ),
    uiOutput("interpretasi_beda_area")
),
tabItem(tabName = "uji_proporsi",
    h2("Uji Proporsi"),
    fluidRow(
        box(width = 4, title = "Pengaturan Uji Proporsi", status = "primary",
            selectInput("var_prop", "Pilih Variabel Kategorik:", choices = var_bebas),
            textInput("nilai_prop1", "Input Nilai (Populasi 1):", value = ""),
            radioButtons("kelompok_prop", "Jumlah Populasi:", choices = c("1 Populasi", "2
Populasi")),
            conditionalPanel(
                condition = "input.kelompok_prop == '2 Populasi'",
                textInput("nilai_prop2", "Input Nilai (Populasi 2):", value = ""),
                selectInput("prov_prop2", "Pilih Provinsi/Kabupaten 2:", choices =
unique(data_raw[[prov_col]])),
                numericInput("p0_2", "Proporsi Hipotesis Populasi 2:", value = 0.5, min = 0, max = 1,

```

```

step = 0.01)
    ),
    selectInput("prov_prop1", "Pilih Provinsi/Kabupaten 1:", choices =
unique(data_raw[[prov_col]])),
    numericInput("p0_1", "Proporsi Hipotesis Populasi 1:", value = 0.5, min = 0, max = 1, step
= 0.01),
    actionButton("uji_prop", "Lakukan Uji Proporsi"),
    tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
    ),
    box(width = 8, title = "Hasil Uji Proporsi", status = "primary",
    verbatimTextOutput("output_prop"),
    uiOutput("interpretasi_prop_area")
    )
    )
),
tabItem(tabName = "uji_varians",
h2("Uji Varians (F-test)",
fluidRow(
box(width = 4, title = "Pengaturan Uji Varians", status = "primary",
selectInput("var_varian", "Pilih Variabel Numerik:", choices = var_numerik),
selectInput("prov_varian1", "Provinsi 1:", choices = unique(data_raw[[prov_col]])),
selectInput("prov_varian2", "Provinsi 2:", choices = unique(data_raw[[prov_col]])),
actionButton("uji_varian", "Lakukan Uji Varians (F-test)",
tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
    ),
    box(width = 8, title = "Hasil Uji Varians", status = "primary",
    verbatimTextOutput("output_varian"),
    uiOutput("interpretasi_varian_area")
    )
    )
),
tabItem(tabName = "anova",

```

```

h2("Uji Beda Rata-rata >2 Kelompok (ANOVA)",
fluidRow(
  box(width = 4, title = "Pengaturan Uji", status = "primary",
    selectInput("var_anova", "Pilih Variabel Respon:", choices = var_numerik),
    pickerInput("prov_anova", "Pilih Minimal 3 Provinsi:", choices =
unique(data_raw[[prov_col]]), multiple = TRUE, options = list('max-options' = 10)),
    actionButton("uji_anova", "Lakukan Uji"),
    tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
  ),
  box(width = 8, title = "Hasil Uji", status = "primary",
    verbatimTextOutput("output_anova"),
    verbatimTextOutput("output_tukey"),
    downloadButton("download_anova", "Unduh Output")
  )
),
uiOutput("interpretasi_anova_area")
),
tabItem(tabName = "regresi",
h2("Regresi Linier Berganda"),
fluidRow(
  box(width = 4, title = "Pengaturan Model", status = "primary",
    selectInput("prov7", "Pilih Provinsi:", choices = c("Semua",
unique(data_raw[[prov_col]]))),
    selectInput("y_regresi", "Variabel Y (Respon):", choices = var_respon),
    pickerInput("x_regresi", "Variabel X (Prediktor):", choices = var_bebas, multiple = TRUE),
    actionButton("run_regresi", "Jalankan Regresi"),
    tags$div(style="color:#0073C2FF;font-size:15px;margin-top:8px;", "Pilih variabel dan/atau
wilayah, lalu hasil/grafik akan muncul.")
  ),
  box(width = 8, title = "Hasil Regresi", status = "primary",
    verbatimTextOutput("output_regresi"),
    downloadButton("download_regresi", "Unduh Output")
  )
)

```

```

    ),
    uiOutput("interpretasi_regresi_area")
  )
)
)
)
)

```

6. Mengatur Fungsi-fungsi Server

```

# Server
server <- function(input, output, session) {
  # --- Manajemen Data ---
  data_kat <- reactiveVal(data_raw)
  observeEvent(input$buat_kat, {
    dat <- data_raw
    if (input$prov != "Semua") {
      dat <- dat[dat[[prov_col]] == input$prov, ]
    }
    var <- input$var_ekonomi
    n_kat <- as.numeric(input$n_kat)
    dat[[paste0(var, "_kat")] <- make_kategori(dat[[var]], n = n_kat)
    data_kat(dat)
    output$interpretasi_kat_area <- renderUI({
      tab <- table(dat[[paste0(var, "_kat")]])
      prop <- round(100 * tab / sum(tab), 2)
      HTML(paste0(
        '<div class="interpretasi-box">',
        'Distribusi kategori untuk variabel <b>', var, '</b> dengan ', n_kat, ' kategori:<br>',
        paste(names(tab), ': ', tab, '(', prop, '%)', collapse = '<br>'),
        '</div>'
      ))
    })
  })
  output$plot_norm <- renderPlot({
    req(input$var_asumsi)
    dat <- data_raw

```



```

if (input$prov3 != "Semua") {
  dat <- dat[dat[[prov_col]] == input$prov3, ]
}
x <- dat[[input$var_asumsi]]
hist(x, breaks = 20, main = paste("Histogram", input$var_asumsi), xlab = input$var_asumsi, col =
"#0073C2FF")
rug(x)
})
output$plot_homog <- renderPlot({
  req(input$var_asumsi)
  provs <- input$prov3_2
  if (!is.null(provs) && length(provs) >= 1) {
    dat2 <- data_raw[data_raw[[prov_col]] %in% c(input$prov3, provs), ]
    x2 <- dat2[[input$var_asumsi]]
    g2 <- dat2[[prov_col]]
    boxplot(x2 ~ g2, main = paste("Boxplot", input$var_asumsi, "per Provinsi"), xlab = "Provinsi", ylab
= input$var_asumsi, col = "#EFC000FF")
  }
})
output$tabel_kat <- renderDT({
  dat <- data_kat()
  var <- input$var_ekonomi
  n_kat <- as.numeric(input$n_kat)
  if (!is.null(dat[[paste0(var, "_kat")]])) {
    dat <- dat[, c(prov_col, kab_col, var, paste0(var, "_kat"))]
  } else {
    dat <- dat[, c(prov_col, kab_col, var)]
  }
  dat
}, options = list(pageLength = 10))
output$download_data <- downloadHandler(
  filename = function() paste0("data_kategori_", input$var_ekonomi, ".xlsx"),
  content = function(file) openxlsx::write.xlsx(data_kat(), file)
)

```

```

output$download_interpretasi_kat <- downloadHandler(
  filename = function() paste0("interpretasi_kategori_", input$var_ekonomi, ".docx"),
  content = function(file) {
    dat <- data_kat()
    var <- input$var_ekonomi
    n_kat <- as.numeric(input$n_kat)
    tab <- table(dat[[paste0(var, "_kat")]])
    prop <- round(100 * tab / sum(tab), 2)
    interpretasi <- paste0(
      "Distribusi kategori untuk variabel ", var, " dengan ", n_kat, " kategori:\n",
      paste(names(tab), ": ", tab, " (", prop, "%)", collapse = "\n")
    )
    doc <- officer::read_docx()
    doc <- officer::body_add_par(doc, interpretasi, style = "Normal")
    print(doc, target = file)
  }
)

output$download_gabungan_kat <- downloadHandler(
  filename = function() paste0("gabungan_kategori_", input$var_ekonomi, ".docx"),
  content = function(file) {
    dat <- data_kat()
    var <- input$var_ekonomi
    n_kat <- as.numeric(input$n_kat)
    tab <- table(dat[[paste0(var, "_kat")]])
    prop <- round(100 * tab / sum(tab), 2)
    interpretasi <- paste0(
      "Distribusi kategori untuk variabel ", var, " dengan ", n_kat, " kategori:\n",
      paste(names(tab), ": ", tab, " (", prop, "%)", collapse = "\n")
    )
    doc <- officer::read_docx()
    doc <- officer::body_add_par(doc, "Tabel Kategori", style = "heading 1")
    doc <- officer::body_add_table(doc, dat[, c(prov_col, kab_col, var, paste0(var, "_kat"))], style =
"table_template")
    doc <- officer::body_add_par(doc, "Interpretasi", style = "heading 1")

```

```

doc <- officer::body_add_par(doc, interpretasi, style = "Normal")
print(doc, target = file)
}
)

# --- Eksplorasi Data ---
plot_eksplorasi_obj <- reactive({
  req(input$var_eksplorasi)
  var <- input$var_eksplorasi
  if (input$prov2 == "Semua") {
    dat <- data_raw %>%
      group_by(.data[[prov_col]], .data[[kdprov_col]]) %>%
      summarise(mean_val = mean(.data[[var]], na.rm = TRUE), .groups = 'drop') %>%
      arrange(.data[[kdprov_col]])
    p <- ggplot2::ggplot(dat, aes(x = reorder(.data[[prov_col]], .data[[kdprov_col]]), y = mean_val)) +
      ggplot2::geom_bar(stat = "identity", fill = "#0073C2FF") +
      ggplot2::labs(x = "Provinsi", y = paste("Rata-rata", var)) +
      ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))
  } else {
    dat <- data_raw %>%
      dplyr::filter(.data[[prov_col]] == input$prov2) %>%
      dplyr::arrange(.data[[kodeprkab_col]])
    p <- ggplot2::ggplot(dat, aes(x = reorder(.data[[kab_col]], .data[[kodeprkab_col]]), y = .data[[var]]))
    +
      ggplot2::geom_bar(stat = "identity", fill = "#EFC000FF") +
      ggplot2::labs(x = "Kabupaten/Kota", y = var) +
      ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 45, hjust = 1))
  }
  p
})
output$plot_eksplorasi <- renderPlotly({
  plotly::ggplotly(plot_eksplorasi_obj())
})
output$download_plot_eksplorasi <- downloadHandler(

```

```

filename = function() paste0("plot_eksplorasi_", input$var_eksplorasi, ".png"),
content = function(file) {
  ggplot2::ggsave(file, plot = plot_eksplorasi_obj(), width = 8, height = 5, dpi = 300)
}
)

# --- Peta Eksplorasi Data ---
output$map_eksplorasi <- renderLeaflet({
  req(input$map_var)
  var <- input$map_var
  geo_data <- geojson
  # Join ke kabupaten/kota jika ada, jika tidak ke provinsi
  if (kodeprkab_col %in% colnames(data_raw) && kodeprkab_col %in% colnames(geo_data)) {
    geo_data <- dplyr::left_join(geo_data, data_raw, by = setNames(kodeprkab_col, kodeprkab_col))
    label_col <- kab_col
  } else if (kdprov_col %in% colnames(data_raw) && kdprov_col %in% colnames(geo_data)) {
    geo_data <- dplyr::left_join(geo_data, data_raw, by = setNames(kdprov_col, kdprov_col))
    label_col <- prov_col
  } else {
    leaflet() %>% addTiles()
    return()
  }
  # Pastikan nilai tidak semuanya NA dan domain valid.
  nilai <- geo_data[[var]]
  if (all(is.na(nilai)) || is.null(nilai)) {
    leaflet() %>% addTiles()
  } else {
    pal <- colorNumeric("YlOrRd", domain = range(nilai, na.rm=TRUE), na.color = "#cccccc")
    labels <- sprintf(
      "<strong>%s</strong><br/>%s: %s",
      geo_data[[label_col]], var, round(nilai, 2)
    ) %>% lapply(htmltools::HTML)
    leaflet(geo_data) %>%
      addTiles() %>%

```

```

addPolygons(
  fillColor = ~pal(nilai),
  weight = 1,
  opacity = 1,
  color = "white",
  dashArray = "3",
  fillOpacity = 0.7,
  highlightOptions = highlightOptions(
    weight = 2,
    color = "#666",
    dashArray = "",
    fillOpacity = 0.7,
    bringToFront = TRUE
  ),
  label = labels,
  labelOptions = labelOptions(
    style = list("font-weight" = "normal", padding = "3px 8px"),
    textSize = "15px",
    direction = "auto"
  )
) %>%
addLegend(
  pal = pal,
  values = nilai,
  opacity = 0.7,
  title = var,
  position = "bottomright"
)
}
})

```

--- Regresi Linier Berganda: Interpretasi Model Terbaik ---

```

observeEvent(input$run_regresi, {
  output$output_regresi <- renderPrint({

```

```

req(input$y_regresi, input$x_regresi)
dat <- data_raw
if (input$prov7 != "Semua") {
  dat <- dat[dat[[prov_col]] == input$prov7, ]
}
formula_str <- paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+"))
model <- lm(as.formula(formula_str), data = dat)
summary(model)
})
output$interpretasi_regresi_area <- renderUI({
  req(input$y_regresi, input$x_regresi)
  dat <- data_raw
  if (input$prov7 != "Semua") {
    dat <- dat[dat[[prov_col]] == input$prov7, ]
  }
  formula_str <- paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+"))
  model <- lm(as.formula(formula_str), data = dat)
  summ <- summary(model)
  coefs <- summ$coefficients
  n_pred <- nrow(coefs) - 1
  interpretasi <- paste0("Model regresi: <b>", formula_str, "</b><br>")
  interpretasi <- paste0(interpretasi, "<ul>")
  for (i in 1:nrow(coefs)) {
    nm <- rownames(coefs)[i]
    if (nm == "(Intercept)") {
      interpretasi <- paste0(interpretasi, "<li>Intercept: Jika semua variabel prediktor bernilai nol, maka
nilai rata-rata respon adalah <b>", round(coefs[i,1],2), "</b>.</li>")
    } else {
      signif <- if (coefs[i,4] < 0.05) "<b>signifikan</b>" else "tidak signifikan"
      interpretasi <- paste0(interpretasi, "<li>Setiap kenaikan 1 satuan <b>", nm, "</b> akan mengubah
<b>", input$y_regresi, "</b> sebesar <b>", round(coefs[i,1],2), "</b> (p-value: ", signif(coefs[i,4],3), ",
", signif, ").</li>")
    }
  }
})

```

```

interpretasi <- paste0(interpretasi, "</ul>")
res <- residuals(model)
norm_res <- shapiro.test(res)
norm_text <- if (norm_res$p.value > 0.05) {
  "Normalitas residual <b>terpenuhi</b>."
} else {
  "Normalitas residual <b>tidak terpenuhi</b>."
}
if (n_pred > 1) {
  vifs <- car::vif(model)
  max_vif <- max(vifs)
  vif_text <- if (max_vif < 10) {
    "Tidak ada multikolinearitas tinggi (VIF < 10)."
```

collapse=", ", ".")

```

  }
} else {
  vif_text <- "Multikolinearitas tidak relevan (hanya 1 prediktor)."
```

Tambahkan interpretasi model terbaik jika lebih dari satu prediktor

```

  }
  bpval <- tryCatch({
    lmtest::bptest(model)$p.value
  }, error=function(e) NA)
  if (!is.na(bpval)) {
    hetero_text <- if (bpval > 0.05) {
      "Tidak ada pelanggaran heteroskedastisitas (uji Breusch-Pagan)."
```

best_model_text <- ""

```

if (length(input$x_regresi) > 1) {
  # Coba semua subset model, pilih yang adj.r.squared tertinggi
  library(leaps)
  regsub <- regsubsets(
    as.formula(paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+"))),
    data = dat, nvmax = min(length(input$x_regresi), 5)
  )
  regsum <- summary(regsub)
  best_adjrsq <- which.max(regsum$adjr2)
  best_vars <- names(which(regsum$outmat[best_adjrsq, ] == "*"))
  best_formula <- paste(input$y_regresi, "~", paste(best_vars, collapse = "+"))
  best_model <- lm(as.formula(best_formula), data = dat)
  best_adjrsq_val <- summary(best_model)$adj.r.squared
  best_model_text <- paste0(
    '<br><b>Model RLB terbaik (adj.R2 tertinggi):</b> ', best_formula,
    '<br>adj.R2 = ', round(best_adjrsq_val, 3),
    '<br>Variabel signifikan: ', paste(names(coef(best_model))[-
1][summary(best_model)$coefficients[-1,4] < 0.05], collapse = ", ")
  )
}

HTML(paste0('<div class="interpretasi-box">',
  interpretasi,
  '<b>Asumsi Klasik:</b><ul>',
  '<li>', norm_text, '</li>',
  '<li>', vif_text, '</li>',
  '<li>', hetero_text, '</li>',
  '</ul>',
  best_model_text,
  '</div>'))
})
})
output$label_eksplorasi <- renderDT({
  req(input$var_eksplorasi)
  var <- input$var_eksplorasi

```



```

if (input$prov2 == "Semua") {
  dat <- data_raw %>%
    group_by(.data[[prov_col]], .data[[kdprov_col]]) %>%
    summarise(mean_val = mean(.data[[var]], na.rm = TRUE), .groups = 'drop') %>%
    arrange(.data[[kdprov_col]])
  dat
} else {
  dat <- data_raw %>%
    filter(.data[[prov_col]] == input$prov2) %>%
    arrange(.data[[kodeprkab_col]])
  dat
}
}, options = list(pageLength = 10))
output$interpretasi_eksplorasi_area <- renderUI({
  req(input$var_eksplorasi)
  var <- input$var_eksplorasi
  if (input$prov2 == "Semua") {
    dat <- data_raw[[var]]
    kat <- make_kategori(dat, n = 3)
    tab <- table(kat)
    mayoritas <- names(tab)[which.max(tab)]
    prop <- round(100 * max(tab) / sum(tab), 1)
    min_val <- min(dat, na.rm=TRUE)
    max_val <- max(dat, na.rm=TRUE)
    daerah_min <- data_raw[[prov_col]][which.min(dat)]
    daerah_max <- data_raw[[prov_col]][which.max(dat)]
    HTML(paste0(
      '<div class="interpretasi-box">',
      'Sebagian besar nilai variabel <b>', var, '</b> pada seluruh provinsi tergolong <b>', mayoritas, '</b>'
      ('', prop, '% dari total data).',
      '<br>Rata-rata: ', round(mean(dat, na.rm=TRUE),2),
      ', Median: ', round(median(dat, na.rm=TRUE),2),
      ', Min: ', round(min_val,2), ' (' , daerah_min, ')',
      ', Max: ', round(max_val,2), ' (' , daerah_max, ')',

```

```

    '</div>'
  ))
} else {
  dat <- data_raw[data_raw[[prov_col]] == input$prov2, var]
  kat <- make_kategori(dat, n = 3)
  tab <- table(kat)
  mayoritas <- names(tab)[which.max(tab)]
  prop <- round(100 * max(tab) / sum(tab), 1)
  min_val <- min(dat, na.rm=TRUE)
  max_val <- max(dat, na.rm=TRUE)
  daerah_min <- data_raw[data_raw[[prov_col]] == input$prov2, kab_col][which.min(dat)]
  daerah_max <- data_raw[data_raw[[prov_col]] == input$prov2, kab_col][which.max(dat)]
  HTML(paste0(
    '<div class="interpretasi-box">',
    'Sebagian besar nilai variabel <b>', var, '</b> di provinsi <b>', input$prov2, '</b> tergolong <b>',
    mayoritas, '</b> (' , prop, '% dari total kabupaten/kota).',
    '<br>Rata-rata: ', round(mean(dat, na.rm=TRUE),2),
    ', Median: ', round(median(dat, na.rm=TRUE),2),
    ', Min: ', round(min_val,2), ' (' , daerah_min, ')',
    ', Max: ', round(max_val,2), ' (' , daerah_max, ')',
    '</div>'
  ))
}
})

# --- Uji Asumsi ---
output$prov3_2_ui <- renderUI({
  prov_choices <- setdiff(unique(data_raw[[prov_col]]), input$prov3)
  pickerInput("prov3_2", "Pilih Provinsi Pembanding (minimal 1):", choices = prov_choices, multiple =
TRUE, options = list('max-options' = 10))
})
output$uji_norm <- renderPrint({
  req(input$var_asisi)
  dat <- data_raw

```

```

if (input$prov3 != "Semua") {
  dat <- dat[dat[[prov_col]] == input$prov3, ]
}
x <- dat[[input$var_asumsi]]
shapiro.test(x)
})
output$uji_homog <- renderPrint({
  req(input$var_asumsi)
  provs <- input$prov3_2
  validate(need(length(provs) >= 1, "Pilih minimal 1 provinsi pembanding untuk uji homogenitas."))
  dat <- data_raw[data_raw[[prov_col]] %in% c(input$prov3, provs), ]
  x <- dat[[input$var_asumsi]]
  g <- dat[[prov_col]]
  car::leveneTest(x, as.factor(g))
})
output$interpretasi_asumsi_area <- renderUI({
  req(input$var_asumsi)
  dat <- data_raw
  if (input$prov3 != "Semua") {
    dat <- dat[dat[[prov_col]] == input$prov3, ]
  }
  x <- dat[[input$var_asumsi]]
  norm_res <- shapiro.test(x)
  norm_text <- if (norm_res$p.value > 0.05) {
    paste0("Data <b>berdistribusi normal</b> (p-value = ", signif(norm_res$p.value,3), ").")
  } else {
    paste0("Data <b>tidak berdistribusi normal</b> (p-value = ", signif(norm_res$p.value,3), ").")
  }
  provs <- input$prov3_2
  homog_text <- ""
  if (!is.null(provs) && length(provs) >= 1) {
    dat2 <- data_raw[data_raw[[prov_col]] %in% c(input$prov3, provs), ]
    x2 <- dat2[[input$var_asumsi]]
    g2 <- dat2[[prov_col]]

```

```

homog_res <- car::leveneTest(x2, as.factor(g2))
pval <- homog_res$`Pr(>F)`[1]
if (pval > 0.05) {
  homog_text <- paste0("Varians antar provinsi <b>homogen</b> (p-value = ", signif(pval,3), ".")
} else {
  homog_text <- paste0("Varians antar provinsi <b>tidak homogen</b> (p-value = ", signif(pval,3),
  ".")
}
}
HTML(paste0('<div class="interpretasi-box">',
  '<b>Normalitas:</b> ', norm_text, '<br>',
  '<b>Homogenitas:</b> ', homog_text,
  '</div>'))
})

```

--- Statistik Inferensia ---

```

observeEvent(input$uji_beda, {
  output$output_beda <- renderPrint({
    req(input$var_beda)
    if (input$kelompok_beda == "1 Provinsi") {
      dat <- data_raw[data_raw[[prov_col]] == input$prov_beda1, ]
      x <- dat[[input$var_beda]]
      mu0 <- input$mu0_beda
      t.test(x, mu = mu0)
    } else {
      dat1 <- data_raw[data_raw[[prov_col]] == input$prov_beda1_2, ]
      dat2 <- data_raw[data_raw[[prov_col]] == input$prov_beda2_2, ]
      t.test(dat1[[input$var_beda]], dat2[[input$var_beda]])
    }
  })
  output$interpretasi_beda_area <- renderUI({
    req(input$var_beda)
    if (input$kelompok_beda == "1 Provinsi") {
      dat <- data_raw[data_raw[[prov_col]] == input$prov_beda1, ]

```

```

x <- dat[[input$var_beda]]
mu0 <- input$mu0_beda
res <- t.test(x, mu = mu0)
kesimpulan <- if (res$p.value < 0.05) {
  "Terdapat perbedaan signifikan antara rata-rata sampel dan hipotesis nol."
} else {
  "Tidak terdapat perbedaan signifikan antara rata-rata sampel dan hipotesis nol."
}
HTML(paste0('<div class="interpretasi-box">',
  'Rata-rata: ', round(mean(x, na.rm=TRUE),2),
  ', Hipotesis nol: ', mu0,
  '<br>p-value: ', signif(res$p.value,3),
  '<br>', kesimpulan,
  '</div>'))
} else {
  dat1 <- data_raw[data_raw[[prov_col]] == input$prov_beda1_2, ]
  dat2 <- data_raw[data_raw[[prov_col]] == input$prov_beda2_2, ]
  res <- t.test(dat1[[input$var_beda]], dat2[[input$var_beda]])
  kesimpulan <- if (res$p.value < 0.05) {
    "Terdapat perbedaan signifikan antara rata-rata kedua provinsi."
  } else {
    "Tidak terdapat perbedaan signifikan antara rata-rata kedua provinsi."
  }
  HTML(paste0('<div class="interpretasi-box">',
    'Rata-rata ', input$prov_beda1_2, ': ', round(mean(dat1[[input$var_beda]], na.rm=TRUE),2),
    ', ', input$prov_beda2_2, ': ', round(mean(dat2[[input$var_beda]], na.rm=TRUE),2),
    '<br>p-value: ', signif(res$p.value,3),
    '<br>', kesimpulan,
    '</div>'))
  }
})
})
observeEvent(input$uji_varian, {
  output$output_varian <- renderPrint({

```

```

req(input$var_varian)
dat1 <- data_raw[data_raw[[prov_col]] == input$prov_varian1, ]
dat2 <- data_raw[data_raw[[prov_col]] == input$prov_varian2, ]
var.test(dat1[[input$var_varian]], dat2[[input$var_varian]])
})
output$interpretasi_varian_area <- renderUI({
  req(input$var_varian)
  dat1 <- data_raw[data_raw[[prov_col]] == input$prov_varian1, ]
  dat2 <- data_raw[data_raw[[prov_col]] == input$prov_varian2, ]
  res <- var.test(dat1[[input$var_varian]], dat2[[input$var_varian]])
  kesimpulan <- if (res$p.value < 0.05) {
    "Terdapat perbedaan signifikan pada varians kedua provinsi."
  } else {
    "Tidak terdapat perbedaan signifikan pada varians kedua provinsi."
  }
  HTML(paste0('<div class="interpretasi-box">',
    'Varians ', input$prov_varian1, ': ', round(var(dat1[[input$var_varian]], na.rm=TRUE),2),
    ', ', input$prov_varian2, ': ', round(var(dat2[[input$var_varian]], na.rm=TRUE),2),
    '<br>p-value: ', signif(res$p.value,3),
    '<br>', kesimpulan,
    '</div>'))
})
})
observeEvent(input$uji_anova, {
  output$output_anova <- renderPrint({
    req(input$var_anova)
    provs <- input$prov_anova
    validate(need(length(provs) >= 3, "Pilih minimal 3 provinsi untuk ANOVA."))
    dat <- data_raw[data_raw[[prov_col]] %in% provs, ]
    aov_res <- aov(dat[[input$var_anova]] ~ as.factor(dat[[prov_col]]))
    summary(aov_res)
  })
  output$output_tukey <- renderPrint({
    req(input$var_anova)

```

```

provs <- input$prov_anova
validate(need(length(provs) >= 3, "Pilih minimal 3 provinsi untuk ANOVA.))
dat <- data_raw[data_raw[[prov_col]] %in% provs, ]
aov_res <- aov(dat[[input$var_anova]] ~ as.factor(dat[[prov_col]]))
TukeyHSD(aov_res)
})
output$interpretasi_anova_area <- renderUI({
  req(input$var_anova)
  provs <- input$prov_anova
  validate(need(length(provs) >= 3, "Pilih minimal 3 provinsi untuk ANOVA.))
  dat <- data_raw[data_raw[[prov_col]] %in% provs, ]
  aov_res <- aov(dat[[input$var_anova]] ~ as.factor(dat[[prov_col]]))
  tukey <- TukeyHSD(aov_res)
  tukey_df <- as.data.frame(tukey[[1]])
  signif_pairs <- rownames(tukey_df)[tukey_df$p adj` < 0.05]
  maxdiff <- if (length(signif_pairs) > 0) {
    signif_pairs[which.max(abs(tukey_df$diff[tukey_df$p adj` < 0.05]))]
  } else {
    NA
  }
  kesimpulan <- if (any(summary(aov_res)[[1]][["Pr(>F)"]][1] < 0.05)) {
    paste0("Terdapat perbedaan rata-rata signifikan antar provinsi. Pasangan provinsi yang paling
berbeda signifikan: <b>", maxdiff, "</b>.")
  } else {
    "Tidak terdapat perbedaan rata-rata signifikan antar provinsi."
  }
  HTML(paste0('<div class="interpretasi-box">',
    'p-value ANOVA: ', signif(summary(aov_res)[[1]][["Pr(>F)"]][1],3),
    '<br>', kesimpulan,
    '</div>'))
})
})
observeEvent(input$uji_prop, {
  output$output_prop <- renderPrint({

```

```

req(input$var_prop)
if (input$kelompok_prop == "1 Populasi") {
  x <- as.numeric(input$nilai_prop1)
  n <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop1, ])
  p0 <- input$p0_1
  prop.test(x, n, p = p0)
} else {
  x1 <- as.numeric(input$nilai_prop1)
  n1 <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop1, ])
  x2 <- as.numeric(input$nilai_prop2)
  n2 <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop2, ])
  p0_1 <- input$p0_1
  p0_2 <- input$p0_2
  prop.test(c(x1, x2), c(n1, n2), p = c(p0_1, p0_2))
}
})
output$interpretasi_prop_area <- renderUI({
  req(input$var_prop)
  if (input$kelompok_prop == "1 Populasi") {
    x <- as.numeric(input$nilai_prop1)
    n <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop1, ])
    p0 <- input$p0_1
    res <- prop.test(x, n, p = p0)
    kesimpulan <- if (res$p.value < 0.05) {
      "Terdapat perbedaan signifikan antara proporsi sampel dan hipotesis nol."
    } else {
      "Tidak terdapat perbedaan signifikan antara proporsi sampel dan hipotesis nol."
    }
  }
  HTML(paste0('<div class="interpretasi-box">',
    'Proporsi sampel: ', round(x/n,3),
    ', Hipotesis nol: ', p0,
    '<br>p-value: ', signif(res$p.value,3),
    '<br>', kesimpulan,
    '</div>'))
})

```



```

} else {
  x1 <- as.numeric(input$nilai_prop1)
  n1 <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop1, ])
  x2 <- as.numeric(input$nilai_prop2)
  n2 <- nrow(data_raw[data_raw[[prov_col]] == input$prov_prop2, ])
  p0_1 <- input$p0_1
  p0_2 <- input$p0_2
  res <- prop.test(c(x1, x2), c(n1, n2), p = c(p0_1, p0_2))
  kesimpulan <- if (res$p.value < 0.05) {
    "Terdapat perbedaan signifikan antara proporsi kedua populasi."
  } else {
    "Tidak terdapat perbedaan signifikan antara proporsi kedua populasi."
  }
  HTML(paste0('<div class="interpretasi-box">',
    'Proporsi 1: ', round(x1/n1,3),
    ', Proporsi 2: ', round(x2/n2,3),
    '<br>p-value: ', signif(res$p.value,3),
    '<br>', kesimpulan,
    '</div>'))
}
})
})

# --- Regresi Linier Berganda ---
observeEvent(input$run_regresi, {
  output$output_regresi <- renderPrint({
    req(input$y_regresi, input$x_regresi)
    dat <- data_raw
    if (input$prov7 != "Semua") {
      dat <- dat[dat[[prov_col]] == input$prov7, ]
    }
    formula_str <- paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+"))
    model <- lm(as.formula(formula_str), data = dat)
    summary(model)
  })
})

```

```

}))
output$interpretasi_regresi_area <- renderUI({
  req(input$y_regresi, input$x_regresi)
  dat <- data_raw
  if (input$prov7 != "Semua") {
    dat <- dat[dat[[prov_col]] == input$prov7, ]
  }
  formula_str <- paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+"))
  model <- lm(as.formula(formula_str), data = dat)
  summ <- summary(model)
  coefs <- summ$coefficients
  n_pred <- nrow(coefs) - 1
  interpretasi <- paste0("Model regresi: <b>", formula_str, "</b><br>")
  interpretasi <- paste0(interpretasi, "<ul>")
  for (i in 1:nrow(coefs)) {
    nm <- rownames(coefs)[i]
    if (nm == "(Intercept)") {
      interpretasi <- paste0(interpretasi, "<li>Intercept: Jika semua variabel prediktor bernilai nol, maka
nilai rata-rata respon adalah <b>", round(coefs[i,1],2), "</b>.</li>")
    } else {
      signif <- if (coefs[i,4] < 0.05) "<b>signifikan</b>" else "tidak signifikan"
      interpretasi <- paste0(interpretasi, "<li>Setiap kenaikan 1 satuan <b>", nm, "</b> akan mengubah
<b>", input$y_regresi, "</b> sebesar <b>", round(coefs[i,1],2), "</b> (p-value: ", signif(coefs[i,4],3), ",
", signif, ").</li>")
    }
  }
  interpretasi <- paste0(interpretasi, "</ul>")
  res <- residuals(model)
  norm_res <- shapiro.test(res)
  norm_text <- if (norm_res$p.value > 0.05) {
    "Normalitas residual <b>terpenuhi</b>."
  } else {
    "Normalitas residual <b>tidak terpenuhi</b>."
  }
})

```

```

if (n_pred > 1) {
  vifs <- car::vif(model)
  max_vif <- max(vifs)
  vif_text <- if (max_vif < 10) {
    "Tidak ada multikolinearitas tinggi (VIF < 10)."
  } else {
    paste0("Terdapat multikolinearitas tinggi pada variabel: ", paste(names(vifs)[vifs >= 10],
collapse=", "), ".")
  }
} else {
  vif_text <- "Multikolinearitas tidak relevan (hanya 1 prediktor)."
}
bpval <- tryCatch({
  lmtest::bptest(model)$p.value
}, error=function(e) NA)
if (!is.na(bpval)) {
  hetero_text <- if (bpval > 0.05) {
    "Tidak ada pelanggaran heteroskedastisitas (uji Breusch-Pagan)."
  } else {
    "Terdapat pelanggaran heteroskedastisitas (uji Breusch-Pagan)."
  }
} else {
  hetero_text <- "Uji heteroskedastisitas tidak tersedia."
}
# Stepwise regression (AIC) untuk model terbaik dari variabel kontrol yang dipilih user
stepwise_text <- ""
if (length(input$x_regresi) > 1) {
  full_formula <- as.formula(paste(input$y_regresi, "~", paste(input$x_regresi, collapse = "+")))
  null_formula <- as.formula(paste(input$y_regresi, "~ 1"))
  dat_step <- data_raw
  if (input$prov7 != "Semua") {
    dat_step <- dat_step[dat_step[[prov_col]] == input$prov7, ]
  }
  suppressWarnings({

```

```

full_model <- lm(full_formula, data = dat_step)
null_model <- lm(null_formula, data = dat_step)
best_model <- step(null_model, scope = list(lower = null_model, upper = full_model), direction =
"both", trace = 0)
})
best_formula <- as.character(formula(best_model))
best_adjrsq <- summary(best_model)$adj.r.squared
signif_vars <- names(coef(best_model))[-1][summary(best_model)$coefficients[-1,4] < 0.05]
stepwise_text <- paste0(
  '<br><b>Model RLB Terbaik dari variabel yang dipilih (Stepwise AIC):</b> ', best_formula[2], ' ~
', best_formula[3],
  '<br>adj.R2 = ', round(best_adjrsq, 3),
  if (length(signif_vars) > 0) paste0('<br>Variabel signifikan: ', paste(signif_vars, collapse=", "))
else "<br>Tidak ada variabel signifikan."
)
}
HTML(paste0('<div class="interpretasi-box">',
  interpretasi,
  '<b>Asumsi Klasik:</b><ul>',
  '<li>', norm_text, '</li>',
  '<li>', vif_text, '</li>',
  '<li>', hetero_text, '</li>',
  '</ul>',
  '<div style="margin-top:16px;border-top:1px solid #ccc;padding-top:8px;">',
  stepwise_text,
  '</div></div>'))
})
})
}

```

shinyApp(ui, server)

Kesimpulan

Ketimpangan sosial ekonomi di Indonesia masih menjadi tantangan besar, terlihat dari variasi indikator kemiskinan, pendidikan, infrastruktur, dan kerentanan bencana antar wilayah. Analisis spasial dan statistik dalam dasbor ini menunjukkan bahwa wilayah dengan akses pendidikan rendah, infrastruktur terbatas, dan kerentanan tinggi cenderung memiliki tingkat kemiskinan yang lebih besar. Dengan pendekatan visual dan interaktif, “Inspiscity” membantu mengidentifikasi wilayah prioritas dan faktor utama penyebab ketimpangan, sehingga dapat menjadi dasar pengambilan kebijakan yang lebih adil dan efektif. Dasbor ini juga membuktikan bahwa integrasi data spasial dan statistik sangat penting untuk memahami isu sosial ekonomi secara komprehensif di era digital.

Penutup

Dengan mengangkat masalah ketimpangan yang masih berlangsung di Indonesia, dasbor “Inspiscity” dapat menjadi suatu alat untuk memudahkan eksplorasi dan analisis ketimpangan sosial ekonomi di Indonesia secara interaktif dan informatif. Dengan fitur visualisasi, uji statistik, dan interpretasi otomatis, pengguna dapat memahami pola ketimpangan dan faktor-faktor yang memengaruhinya secara lebih mendalam. Dasbor ini diharapkan dapat menjadi rujukan bagi pengambil kebijakan, peneliti, maupun masyarakat umum yang peduli pada isu ketimpangan sosial ekonomi di Indonesia.